



CINTED-UFRGS

Novas Tecnologias na Educação

UNIVERSIDADE FEDERAL DO RIO GRANDE DO SUL

CINTED - Centro Interdisciplinar de Novas Tecnologias na Educação

Ciclo de Palestras: Inovações em Tecnologia na Educação: Processos e Produtos

**INTEGRANDO CONHECIMENTOS, APROXIMANDO DISCIPLINAS: A  
IMPORTÂNCIA DO *DESIGN* E DA ERGONOMIA NO PROJETO E NO  
DESENVOLVIMENTO DE *SOFTWARES* EDUCACIONAIS.**

**Gabriela Trindade Perry** - (UFRGS) - [trindade@producao.ufrgs.br](mailto:trindade@producao.ufrgs.br)  
**Agostinho Serrano Andrade Neto** – (ULBRA) – [serrano@ulbra.tche.br](mailto:serrano@ulbra.tche.br)  
**Fernando Gonçalves Amaral** – (UFRGS)– [amaral@producao.ufrgs.br](mailto:amaral@producao.ufrgs.br)

Porto Alegre, Março de 2004



## ***Integrando conhecimentos, aproximando disciplinas: a importância do Design e da Ergonomia no projeto e no desenvolvimento de softwares educacionais.***

Gabriela Trindade Perry\*  
Agostinho Serrano Andrade Neto\*\*  
Fernando Gonçalves Amaral\*\*\*

### **Resumo**

O objetivo deste artigo é apresentar à comunidade que desenvolve *softwares* educacionais algumas técnicas e princípios ergonômicos e de *design*, que têm a importante função de integrar os conhecimentos dos diversos membros da equipe e dos usuários. É sustentado que o papel do *design* não se resume à especificação gráfica dos componentes da interface, da mesma forma que a Ergonomia não se resume ao emprego de checklists. Desta forma, será feito um panorama de técnicas e procedimentos, organizados em relação ao momento indicado ao uso: projeto, implementação e avaliação.

Palavras-chave: *design* de interface, avaliação formativa de interface, técnicas de *design*, ergonomia, e interação homem-computador.

### **Abstract**

*The aim of this paper is to present to the community who develops educational software some principles and techniques concerning Ergonomics and Design, which have the important function on integrating the members' knowledge of the team and users. It is believed that the role of design is not restrict to the specification of interface's graphics components, as like as Ergonomics is not restrict to the use of checklists. So, a panorama of techniques and procedures will be shown, organized according to the time for use: project, implementation and evaluation.*

*Keywords: interface design, formative evaluation of interface, design techniques, ergonomics and human-computer interaction.*

### **1. Introdução**

O eixo da discussão deste artigo envolve dois aspectos críticos da produção de *softwares* educacionais: a integração entre profissionais da educação e informatas e a sistematização do desenvolvimento do *software*. Estas duas questões se complementam, pois, para integrar a equipe, é necessário prever quando e de forma acontecerão as interações, ou seja, sistematizá-las. E este é um problema sobre o qual se debruçam as metodologias de desenvolvimento de *software*. Elas costumam apresentar as fases projeto instrucional, desenvolvimento e avaliação, cada uma com diversos desdobramentos. Outra tendência é a adoção de ciclos baseados em prototipagem evolutiva, que prevêem construção de protótipos através de múltiplas iterações, ocasionadas pela agregação de novos requisitos ao projeto. Também todas concordam que tais projetos devem ser conduzidos por equipes multidisciplinares. Exemplos significativos e bem estruturados são os trabalhos de Batista (1997); Burd (1999);

\* Estudante de Mestrado Acadêmico – Bacharel em *Design* – Programa de Pós-Graduação em Engenharia de Produção – PPGEP/UFRGS – [trindade@producao.ufrgs.br](mailto:trindade@producao.ufrgs.br) – <http://www.gabriela.trindade.nom.br>

\*\* Professor adjunto - Doutor em Física – Programa de Pós-Graduação em Ensino de Ciências e Matemática – [serrano@ulbra.tche.br](mailto:serrano@ulbra.tche.br) – <http://www.ulbra.br>

\*\*\* Professor adjunto - Doutor em *Sciences de la Santé Publique*, orientação Ergonomia - Programa de Pós-Graduação em Engenharia de Produção – PPGEP/UFRGS – [amaral@producao.ufrgs.br](mailto:amaral@producao.ufrgs.br) – <http://www.producao.ufrgs.br>

Crosier *et al* (2002); Van der Mast (1995) e Pernin (1996). Porém, o que se deseja focar neste artigo, é o conjunto de algumas conclusões extraídas da revisão destas propostas.

- O gerenciamento de uma equipe com tal caráter multidisciplinar é difícil. O gerente de uma equipe como esta deve ter condições de dialogar com todos os outros membros.
- A comunicação entre os membros desta equipe é difícil.
- Medidas e avaliações de qualidade são necessárias.
- O projeto da interface deve ser considerado prioritário.

Estas constatações apontam a necessidade de favorecer a integração da equipe, dentro de uma perspectiva de desenvolvimento sistematizada. No entanto, é preciso tomar alguns cuidados. Inserir os usuários - neste caso, os professores e os estudantes – no processo não é uma saída mágica. Bjerknes (2003) dá algumas sugestões para uma inserção eficiente e efetiva. Um dos pontos de partida é dar-se conta de que, mesmo que usuários e desenvolvedores estejam dispostos a cooperarem, estas atividades interferem em seu trabalho, em seus compromissos. Por isso, deve ser especificado no contrato quanto tempo será disponibilizado para a participação. Incluir pessoas com poder de decisão e habilidade para intervir caso seja necessário (como em casos de tensões não resolvidas). Em relação ao envolvimento de usuários no processo de desenvolvimento da interface, Rubin (1994) destaca que isto pode trazer complicações, pois a equipe de desenvolvimento talvez não saiba interpretar as reações dos usuários. Em relação à iteratividade, Myers (1994) adverte que a intuição de um *designer* de como consertar um problema observado pode estar errada, de modo que a versão nova pode ficar pior que a anterior. Finalmente, ciclos iterativos podem ser demorados e caros. Bailey (1993) afirma que dados experimentais dão suporte à idéia que mudanças introduzidas para resolver um problema de usabilidade acabam gerando outro problema. O mesmo artigo indica que iterar um *design* ruim melhora-o bastante, mas este método não faz com que o *design* fique tão bom quanto ele seria se tivesse sido originalmente bem projetado. Resumindo, a iteração não substitui bons projetos.

Por estes motivos, a contribuição que há tanto tempo vem sendo feita pela Ergonomia e pelo *Design*, é apresentada de acordo com o melhor momento para sua aplicação: o projeto, a implementação ou a avaliação.

## 2. Atividades de Projeto

Uma das atividades de projeto mais importante é, certamente, a análise de requisitos. Uma análise bem feita evita custos e retrabalho, aumentando a velocidade e a qualidade do produto. Contudo, existe uma imensa dificuldade para realizar uma análise eficiente: os desenvolvedores não conhecem a tarefa executada pelo usuário. Outra questão é que, em situações “reais”, novos requisitos vão sendo incorporados ao *software*, pois também é difícil, para o cliente, saber como deve ser o produto que ele imagina. Este é, por sinal, o motor de algumas novas filosofias de desenvolvimento de *software*, os *agile processes*<sup>1</sup>. Então, será apresentada uma abordagem tipicamente associada às metodologias ágeis, chamada JAD (*Joint Application Development*). Em seguida, será mostrada uma proposta originada no PPGE<sup>2</sup>-UFRGS, chamada DM - *Design MacroErgonômico*.

---

<sup>1</sup> Para mais detalhes ver Beck (2001).

<sup>2</sup> PPGE – Programa de Pós Graduação em Engenharia de Produção  
V. 3 Nº 1, Maio, 2005

### 2.1. JAD – Joint Application Development<sup>3</sup>

JAD é um processo de grupo onde os participantes interagem livremente, o que substitui a técnica de empregar entrevistas com usuários para determinar requisitos de um sistema. Duggan & Thachenkary (2004) afirmam que JAD é considerada *best practice* para aumentar o comprometimento com o usuário, e um investimento que reduz riscos associados ao desenvolvimento de *software*. Segundo o autor, participam das reuniões: um facilitador; usuários, gerentes e desenvolvedores; um secretário e um observador. Uma sessão JAD tem cinco fases: definição do tema; pesquisa; preparação; reunião; elaboração do documento final.

### 2.2. DM – Design MacroErgonômico

Uma ferramenta eficiente, porém que parece nunca ter sido empregada em contextos de engenharia de *software* é o *Design Macroergonômico* – DM - proposto por Fogliatto e Guimarães (1999). Esta ferramenta foi, segundo os autores, projetada para incorporar as necessidades dos usuários em produtos e postos de trabalho. A implementação do DM contempla as seguintes etapas.

1. Identificação do usuário e coleta organizada de informações acerca de sua demanda ergonômica através de entrevistas abertas e questionários
2. Priorização dos itens de demanda ergonômica (IDE's) com os usuários.
3. Incorporação da opinião de especialistas.
4. Listagem, pelos especialistas, dos itens de *design* (ID's) a serem considerados.
5. Determinação da relação de força entre os IDE's e os ID's. O objetivo é identificar grupos de ID's a priorizar.
6. Tratamento ergonômicos dos ID's.
7. Implementação do novo *design* e acompanhamento.

O resultado deste processo, das etapas 1 à 5, é uma matriz onde estão representados os itens a que se deve dar maior atenção, por causa de seu impacto na tarefa.

Ambas são ferramentas bastante úteis para reunir as necessidades de todos os envolvidos no projeto.

### 3. Atividades de Implementação

Nesta seção serão expostas técnicas que permitem que todos participem da elaboração do *design* da interface, pois não necessitam (necessariamente), de um programador para criar protótipos. São de um modo geral, simples, rápidas e fáceis de serem conduzidas.

#### 3.1. PICTIVE – Plastic Interface for Collaborative Technology Initiatives through Video Exploration

Apresentada por Muller (1991), combina artefatos com pouca tecnologia e gravações em vídeo. Os primeiros representam funcionalidades do sistema, e por serem de simples produção e operação, permite que todos contribuam com idéias. Estes materiais se dividem em duas categorias: de escritório e os preparados pela equipe de desenvolvimento (ícones, diálogos etc.). A idéia é que o *mock-up* produzido seja modificado pelos usuários em tempo real, como peças de um quebra-cabeça. O vídeo é

<sup>3</sup> Alguns sinônimos para JAD são: *Joint Application Design / Development*; *Interactive Design*; *Group Design*; *Accelerated Design*; *Team Analysis*.

a “documentação” do processo. O autor ressalta ainda que é adequado utilizar PICTIVE em projetos que tenham disponíveis usuários com conhecimento da tarefa. Por esta razão, não aconselha seu uso em projetos de desenvolvimento de novas tecnologias.

### 3.2. CARD – Collaborative Analysis of Requirements and Design

É uma técnica participativa que permite analisar e redesenhar o fluxo da tarefa em sistemas computacionais, através de jogos de cartão, informais e semi-estruturados. Cada ícone ou componente da tela é impresso em um cartão, e enquanto o usuário descreve uma determinada tarefa, o facilitador mostra o cartão relacionado. Em seguida, o usuário é convidado a expressar suas idéias para melhorar algum aspecto do sistema em estudo através dos cartões. Assim, ele representa um fluxo melhorado da tarefa, mostrando um cartão após o outro, sendo o procedimento filmado e analisado posteriormente (MULLER 2001). A Figura 1 mostra um exemplo retirado da coluna *Alertbox*, de Nielsen (2004).

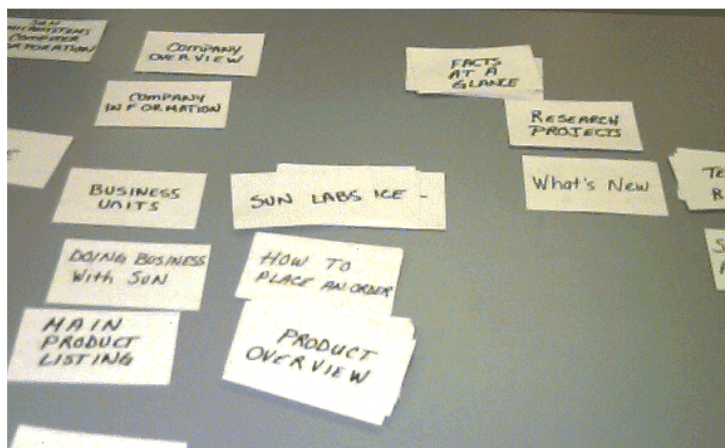


Figura 1. Design da navegação da intranet da Sun através de um jogo com cartões. Extraído de Nielsen (2004).

## 4. Atividades de Avaliação

Este é um dos temas mais caros à pesquisa em Interação Homem-Computador, pois é o momento em que a interface é posta à prova. A quantidade de informações disponível é muito grande, e as técnicas são as mais diversas. A seguir serão mostradas abordagens para avaliação da interface sem usuários e com usuários. O grupo das quatro primeiras é maior, pois são mais rápidas e baratas de serem feitas. Usualmente deixa-se para realizar testes com usuários no final do projeto, por causa do custo associado a este procedimento.

### 4.1. Modelagem da Performance dos Usuários – GOMS

Segundo Hochstein (2002), GOMS (*Goals, Operators, Methods and Selection Rules*) visa analisar a complexidade da interação do usuário com a interface. Resumidamente, consiste no uso de métodos empregados para atingir certo objetivo. Um método é uma seqüência de operadores. Se há mais de um método para atingir um objetivo, são aplicadas regras de seleção. A Figura 2 traz um exemplo do uso de GOMS para a tarefa de editar um texto em um processador.



GOAL: EDIT-MANUSCRIPT			
. GOAL: EDIT-UNIT-TASK ... repeat until no more unit tasks			
. . GOAL: ACQUIRE UNIT-TASK			
. . . GOAL: GET-NEXT-PAGE ... if at end of manuscript page			
. . . GOAL: GET-FROM-MANUSCRIPT			
. . GOAL: EXECUTE-UNIT-TASK ... if a unit task was found			
. . . GOAL: MODIFY-TEXT			
. . . . [select: GOAL: MOVE-TEXT* ...if text is to be moved			
. . . . . GOAL: DELETE-PHRASE ...if a phrase is to be deleted			
. . . . . GOAL: INSERT-WORD] ... if a word is to be inserted			
. . . . VERIFY-EDIT			
*Expansion of MOVE-TEXT goal			
GOAL: MOVE-TEXT			
. GOAL: CUT-TEXT			
. . GOAL: HIGHLIGHT-TEXT			
. . . [select**: GOAL: HIGHLIGHT-WORD			
. . . . MOVE-CURSOR-TO-WORD			
. . . . DOUBLE-CLICK-MOUSE-BUTTON			
. . . . VERIFY-HIGHLIGHT			
. . . GOAL: HIGHLIGHT-ARBITRARY-TEXT			
. . . . MOVE-CURSOR-TO-BEGINNING	1.10		
. . . . CLICK-MOUSE-BUTTON	0.20		
. . . . MOVE-CURSOR-TO-END	1.10		
. . . . SHIFT-CLICK-MOUSE-BUTTON	0.48		
. . . . VERIFY-HIGHLIGHT]	1.35		
. . GOAL: ISSUE-CUT-COMMAND			
. . . MOVE-CURSOR-TO-EDIT-MENU		1.10	
. . . PRESS-MOUSE-BUTTON	0.10		
. . . MOVE-CURSOR-TO-CUT-ITEM	1.10		
. . . VERIFY-HIGHLIGHT	1.35		
. . . RELEASE-MOUSE-BUTTON	0.10		
. GOAL: PASTE-TEXT			
. . GOAL: POSITION-CURSOR-AT-INSERTION-POINT			
. . . MOVE-CURSOR-TO-INSERTION-POINT	1.10		
. . . CLICK-MOUSE-BUTTON	0.20		
. . . VERIFY-POSITION		1.35	
. . GOAL: ISSUE-PASTE-COMMAND			
. . . MOVE-CURSOR-TO-EDIT-MENU		1.10	
. . . PRESS-MOUSE-BUTTON	0.10		
. . . MOVE-MOUSE-TO-PASTE-ITEM		1.10	
. . . VERIFY-HIGHLIGHT	1.35		
. . . RELEASE-MOUSE-BUTTON	0.10		
TOTAL TIME PREDICTED (SEC)			14.38

Figura 2. Baseado na análise GOMS apresentada, esta tarefa deve levar 14.38 segundos para ser realizada. Extraída de Hochstein (2002).

## 4.2. Guidelines

Guidelines são recomendações. Isto implica que cabe ao *designer* escolher segui-las. Na literatura, podem ser encontradas quer sob a forma de recomendações gerais (por exemplo: agrupe componentes que executam funções parecidas) ou que tratam de algum item específico (por exemplo: guidelines para *Help*, para entrada de dados, etc.). Para não alongar esta apresentação, decidiu-se apresentar os Critérios Ergonômicos de Bastien & Scapin (1993), um relatório de pesquisa que apresenta oito critérios ergonômicos para avaliação de interfaces. Os critérios apresentados são: orientação; carga de trabalho; controle explícito; adaptabilidade; gerenciamento de erros; consistência; significado dos códigos e compatibilidade. Alguns destes critérios estão divididos em sub-critérios, sendo cada um deles apresentado com sua descrição, justificativa, exemplos de guidelines e comentários. Para favorecer a compreensão da



organização e do uso desta pesquisa, transcreveu-se resumidamente na Figura 3 o sub-critério “Legibilidade”, parte do critério “Orientação”.

---

#### LEGIBILIDADE

Definição. Relaciona-se com as características da tela que podem favorecer ou dificultar a leitura da informação.

Justificativa. As performances são melhoradas quando a apresentação da informação na tela considera as características cognitivas e perceptuais do usuário.

Exemplos de Guidelines. Títulos devem ser centralizados; rótulos devem ser mostrados em maiúsculas.

---

Figura 3. Exemplo de organização e uso dos Critérios Ergonômicos de Bastien & Scapin (1993).

### 4.3. Checklists

São verificações que podem ser conduzidas por não especialistas em Interação Homem-Computador, que diagnosticam problemas gerais da interface. A qualidade da verificação está diretamente relacionada à da lista. Segundo Cybis (2003) a avaliação através de checklists apresenta as características seguintes.

- Não exige especialistas em Interação Homem-Computador.
- Sistematização da avaliação, o que garante a consistência dos resultados mesmo quando a ferramenta é aplicada por diferentes avaliadores.
- Facilidade na identificação dos problemas de usabilidade, devido à especificidade das questões.
- Redução da subjetividade.
- Redução do custo.

O ErgoList (<http://www.labiutil.inf.ufsc.br/ergolist>) é um exemplo deste tipo de ferramenta.

### 4.4. Análise Heurística

Molich & Nielsen (1990) realizaram uma pesquisa com 77 *designers* e programadores, da indústria e acadêmicos, para investigar se conseguiam encontrar problemas de usabilidade em uma interface. A hipótese dos autores é que as guidelines não orientam o *design* de maneira adequada, por não serem específicas e porque os manuais são muito grandes. Como solução, propõe nove heurísticas de usabilidade (Figura 4), identificadas pela experiência dos autores, e que, segundo os mesmos, estavam “implícitas ou explícitas na maioria das listas sugeridas para IHC”.

- 
1. **Diálogos devem ser simples e naturais.** Simples significa não ter informações irrelevantes. Natural significa uma ordem que faça sentido no contexto da tarefa.
  2. **Fale a linguagem do usuário.** Use palavras e conceitos compreensíveis, que façam parte do universo do usuário. Não use termos específicos ou de engenharia.
  3. **Minimize a carga de memória do usuário.** Não o force a lembrar de coisas de uma ação para outra. Deixe a informação na tela até que ela não seja mais necessária.
  4. **Seja consistente.** Os usuários devem ser capazes de aprender uma sequência de ações em uma parte do sistema e aplicá-la em outra.
  5. **Forneça retorno (feedback).** Os usuários devem saber o resultado de suas ações.
  6. **Forneça saídas claras.** Se o usuário entrar numa parte do sistema que não o interessa, ele deve conseguir sair rapidamente, sem causar danos.
  7. **Forneça atalhos.** Atalhos ajudam usuários experientes.
  8. **Boas mensagens de erro.** Permitem que o usuário saiba qual o problema e como corrigi-lo.
  9. **Prevenção de erros.** Quando se escreve uma mensagem de erro deve-se perguntar se ele não pode ser evitado.
-

Figura 4. As nove heurísticas de Molich &amp; Nielsen (1990).

#### 4.5. Avaliação com Usuários

O estudo dos testes com o usuário é um assunto de grande importância para a IHC. Rubin (1994), por exemplo, detalha diversos tipos de testes com usuários. Em seu livro encontram-se desde tutoriais passo a passo para realização de cada deles, o momento de utilizá-los e sugestões para configuração do ambiente (sala, laboratório) onde será feito. O autor faz algumas considerações em relação à escolha de participantes, conforme a lista a seguir.

- Documente o teste e defina os termos com precisão.
- Escolha o número de participantes de acordo com o grau de confiança desejado para o teste, a quantidade de recursos disponíveis, a disponibilidade do tipo de participante desejado, o tempo disponível para prepara e rodar o teste.
- Inclua pelo menos alguns participantes menos competentes-hábeis-experientes.
- Cuidados ao usar participantes internos, membros da equipe ou funcionários.
- Cuidado para não testar apenas os mais hábeis.

#### 5. Conclusões

Com esta exposição, pretendeu-se mostrar a diversidade de meios para promover a integração entre os membros de um time de desenvolvimento. É certo que as técnicas elencadas acima não foram motivadas por *softwares* educacionais, todavia, julga-se que elas se aplicam.

#### 6. Referências Bibliográficas

BAILEY, Greg. Iterative Methodology and *Designer* Training in Human Computer Interface *Design*. In: **Conference on Human Factors in Computing Systems**. 1993, Amsterdam. Anais... p. 198-205.

BASTIEN, J. M. C.; SCAPIN, D. L.: **Ergonomic criteria for the evaluation of human-computer interfaces**. Relatório de Pesquisa n° 0156. INRIA – Institut National de Recherche en Informatique et en Automatique. Rocquencourt, França, 1993. Disponível em: <<http://www.inria.fr/rrrt/rt-0156.html>>. Acesso em: 15 mar 2004.

BATISTA, J.; FIGUEIREDO, A. D.: Desenvolvimento de programas educativos por prototipagem continuamente evolutiva. In: **2º Simpósio Investigação e Desenvolvimento de Software Educativo**, 1997, Coimbra. Proceedings. Coimbra, Portugal, 1997. Disponível em: <<http://lsm.dei.uc.pt/simposio/pdfs/c07.pdf>>. Acesso em 10 nov. 2004.

BECK, K. *et al.* **Manifesto for agile software development**, 2001. Disponível em: <<http://agilemanifesto.org>>. Acesso em 10 de jun. 2004.



BJERKNER, G.: Some PD advice. **Communications of the ACM**. 36(4), p. 39, 1993.

BURD, L.. **Desenvolvimento de softwares para atividades educacionais**. Campinas: UNICAMP, 1999. Dissertação, Universidade de Campinas, 1999.

CROSIER, J. K.; COBB, S.; WILSON, J. R.: Key lessons for the *design* and integration of virtual environments in secondary science. **Computers & Education**, 38, pp. 77-94, 2002.

CYBIS, Walter de Abreu. **Engenharia de Usabilidade: Uma Visão Ergonômica**. Apostila, Maio 2003. Florianópolis, Maio 2003. Disponível em: <[www.labiutil.inf.ufsc.br/Apostila\\_nvVersao.pdf](http://www.labiutil.inf.ufsc.br/Apostila_nvVersao.pdf)>. Acesso em: 05 Mar 2004.

DUGGAN, E. W.; THACHENKARY, C. S.: Integrating nominal group technique and joint application development for improved systems requirements determination. **Information & Management**, 41, pp. 399-411. 2004.

FOGLIATTO, F. S.; GUIMARÃES, L. B. M.: *Design* macroergonômico: uma proposta metodológica para projeto de produto. **Produto & Produção**, 3 (3), 1999.

HOCHSTEIN, L. **GOMS**, 2002. Disponível em: <<http://www.cs.umd.edu/class/fall2002/cmsc838s/tichi/goms.html>>. Acesso em 10 abr 2004.

MOLICH, R.; NIELSEN, J.: Improving a human-computer dialogue: What *designers* know about traditional interface *design*. **Communications of the ACM**, 33, pp. 338-342, 1990.

MULLER, M. J.: Layered participatory analysis: new developments in the CARD technique. In: **CHI'01 - Conference on Human Factors in Computer Systems**. New York: ACM, 2001, pp. 90-97.

\_\_\_\_\_: PICTIVE: an exploration in participatory *design*. In: **CHI'91 - Conference on Human Factors in Computer Systems**. New Orleans: ACM, 1991, p. 225-231.

MYERS, Brad. A. Challenges of HCI *Design* and Implementation. **Interactions**. Jan. 1994.

NIELSEN, J.: **Card sorting: How many users to test**. Alertbox, 19 Julho, 2004. Disponível em: <<http://www.useit.com/alertbox/20040719.html>>. Acesso em: 10 set. 2004.



PERNIN, Jean-Philippe. **M.A.R.S. Un Modèle Opérationnel de Conception de Simulations Pédagogiques**. Grenoble : Université Joseph Fourier, 1996. Tese, Laboratoire CLIPS – IMAG, Université Joseph Fourier, 1996.

RUBIN, Jeffrey. **Handbook of usability testing**: how to plan, *design*, and conduct effective tests. New York : John Wiley, 1994.

VAN DER MAST, C. A. P. G.: **Developing educational software: integrating disciplines and media**. Tese. Technische Universiteit Delft, 1995.